

19 BUNDESREPUBLIK

DEUTSCHLAND



DEUTSCHES

PATENTAMT

12 Offenlegungsschrift

10 DE 40 13 851 A 1

51 Int. Cl. 5:

G 09 G 1/00

21 Aktenzeichen: P 40 13 851.8

22 Anmeldetag: 30. 4. 90

43 Offenlegungstag: 28. 11. 91

DE 40 13 851 A 1

71 Anmelder:

Aero-Data-Flugmeßtechnik-GmbH, 3300
Braunschweig, DE

74 Vertreter:

Gramm, W., Prof.Dipl.-Ing.; Lins, E., Dipl.-Phys.,
Pat.-Anwälte, 3300 Braunschweig

72 Erfinder:

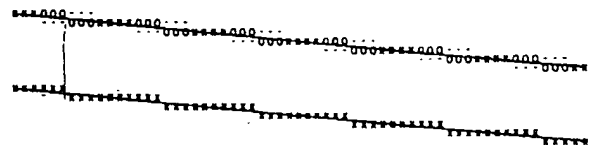
Bavendiek, Klaus, 3300 Braunschweig, DE

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Verfahren zur grafischen Darstellung einer Geraden auf einem Bildschirm

57 Zur grafischen Darstellung einer zwischen einem Anfangspunkt und einem Endpunkt verlaufenden Geraden auf einem Bildschirm unter Verwendung eines die Pixel des Bildschirms ansteuernden Algorithmus, wird ein Verfahren angegeben, das den Eindruck störender Stufen vermindert und die Darstellung von Geraden mit unterschiedlichen Winkeln mit gleicher Helligkeit ermöglicht. Die hierfür durchzuführenden Verfahrensschritte sind

- Ermittlung des Abstandes des Anfangspunktes vom Endpunkt in X- und Y-Richtung
- Unterscheidung verschiedener Winkelbereiche
- für Winkel $> 0^\circ$ gegenüber der Horizontalen, Ersetzen von wenigstens einem normal angesteuerten Pixelpunkt durch eine Mehrzahl von Pixelpunkten mit einer verringerten Intensität in für die unterschiedenen Winkelbereiche unterschiedlichen Konfigurationen



XS= 1 YS= 1 KE= 62 YE= 7 DE= 61 DT= 6
XS= 1 YS= 1 KE= 62 YE= 7 DT= 1
DE= 61 DT= 6 E= -1 DT= 12 -50

DE 40 13 851 A 1

Beschreibung

Die Erfindung betrifft ein Verfahren zur grafischen Darstellung einer zwischen einem Anfangspunkt und einem Endpunkt verlaufenden Geraden auf einem Bildschirm unter Verwendung eines die Pixel des Bildschirms ansteuernden Algorithmus.

Für die Darstellung von Geraden auf einem Bildschirm müssen naturgemäß die regelmäßig in Zeilen und Spalten angeordneten Pixel verwendet werden. Sind die Geraden gegenüber der Horizontalen bzw. der Vertikalen geneigt, müssen die Pixel Stufen bilden, um resultierend die Gerade darzustellen. Insbesondere bei geringen Abweichungen von den Winkeln 0° , 45° und 90° treten erhebliche, deutlich sichtbare Stufen auf, die in manchen Anwendungsfällen besonders störend sind.

Die Berechnung der anzusteuern Pixel des Bildschirms erfolgt üblicherweise mit dem Bresenham-Algorithmus, der auf Multiplikationen und Divisionen verzichtet und daher relativ schnell arbeitet. Nachteilig an dem Bresenham-Algorithmus sind die deutlich sichtbaren Treppenstufen der gezogenen Linien sowie die Tatsache, daß eine 45° -Linie eine geringere Helligkeit als eine 0° -Linie hat, weil die Pixeldichte auf der 45° -Linie um den Faktor $\sqrt{2}$ kleiner ist als auf der 0° -Linie.

Es ist bereits vorgeschlagen worden, den störenden Eindruck der Stufen in den gezogenen Linien dadurch zu vermindern, daß in den Stufen zusätzliche Punkte mit geringerer Intensität gesetzt werden. Hierfür sind zeitintensive Software-Lösungen vorgeschlagen worden, bei denen für jede Linie die Steigung berechnet und mit Hilfe der Nachkomma-Stellen die Punkte im Raster für die zusätzlichen Pixel mit geringerer Intensität ermittelt wurden. Für das Ziehen einer Linie ist dabei etwa die doppelte Zeit gegenüber der Linie nach dem Bresenham-Algorithmus benötigt worden. Eine derartige Lösung ist daher nachteilig.

Das der Erfindung zugrundeliegende Problem besteht somit darin, die Qualität der grafischen Darstellung einer Geraden auf einem Bildschirm zu verbessern.

Ausgehend von dieser Problemstellung ist ein Verfahren der eingangs erwähnten Art erfindungsgemäß durch folgende Verfahrensschritte gekennzeichnet:

- Ermittlung des Abstandes des Anfangspunktes vom Endpunkt in X- und Y-Richtung
- Unterscheidung verschiedener Winkelbereiche
- für Winkel > 0 gegenüber der Horizontalen Ersetzen von wenigstens einem normal angesteuerten Pixelpunkt durch eine Mehrzahl von Pixelpunkten mit einer verringerten Intensität in für die unterschiedlichen Winkelbereiche unterschiedlichen Konfigurationen.

Das erfindungsgemäße Verfahren geht im Unterschied zum Bresenham-Algorithmus nicht davon aus, daß Einzelpixel linear (unter Bildung der Stufen) aneinandergereiht werden, um so eine der dargestellten Gerade angenäherte Gerade zu bilden. Im Gegensatz dazu sieht das erfindungsgemäße Verfahren vor, in vielen Fällen solche Einzelpixel durch Pixelpaare zu ersetzen, die mit verringerter Intensität angesteuert werden, und zwar auch für den Fall, daß die gewünschte Gerade genau durch ein Pixel hindurchläuft. So ist beispielsweise in einem bevorzugten Ausführungsbeispiel der Erfindung die Darstellung einer 45° -Linie, die ausschließlich (ohne Bildung von Stufen) durch Pixel hindurchläuft, ausschließlich mit Pixelpaaren erfolgt, wobei der Anfangs- und der Endpunkt ausgenommen sind. Kennzeichnend für die vorliegende Erfindung sind daher Pixelkonfigurationen mit Pixelpunkten verringerter Intensität, die normal angesteuerte Pixelpunkte – vorzugsweise im Bereich von sonst entstehenden Stufen – ersetzen. Für die erwähnte 45° -Linie führt die Erfindung dazu, daß die Helligkeit dieser Linie genauso groß sein kann, wie die Helligkeit einer 0° -Linie. In einer bevorzugten Ausführungsform der Erfindung erfolgt die Ersetzung der normal angesteuerten Pixelpunkte mit Konfigurationen von Pixelpunkten mit verringerter Intensität unter dem Gesichtspunkt der gleichen Helligkeit für unter verschiedenen Winkeln dargestellte Geraden.

Zweckmäßig ist eine Ausgestaltung der Erfindung, bei der die verringerte Intensität 65% – 80% der normalen Intensität beträgt. Bevorzugt ist allerdings die Verwendung von zwei unterschiedlich verringerten Intensitäten, von denen die erste verringerte Intensität 65% – 80% der normalen Intensität und die zweite verringerte Intensität 45% – 60% der normalen Intensität beträgt. Bei Farbdarstellungen wird die jeweils verringerte Intensität farbabhängig aus einer üblichen Farbtabelle ausgewählt.

Ein sehr funktionstüchtiger und schneller Algorithmus läßt sich erstellen, wenn eine Unterscheidung in wenigstens drei, vorzugsweise fünf Winkelbereiche zwischen 0° und 45° vorgenommen wird, wobei zweckmäßigerweise die Winkel 0° und 45° gesondert behandelt werden. Für die Schnelligkeit ist es von Bedeutung, daß der Algorithmus ohne Multiplikationen und Divisionen auskommt. Dies gilt auch für die Aufteilung in die verschiedenen Winkelbereiche, die vorzugsweise durch Additionen und Subtraktionen der Abstände von Anfangs- und Endpunkt in X- und Y-Richtung bzw. hieraus durch Addition und/oder Subtraktion gebildeter Hilfsgrößen erfolgt.

Für die durch den erfindungsgemäßen Algorithmus erreichbare Bearbeitungsgeschwindigkeit ist es wesentlich, daß bei der Aufteilung in die unterschiedlichen Winkelbereiche Gesetzmäßigkeiten vorliegen, die es erlauben, bestimmte Pixelpunkte bzw. Konfigurationen beim Vorliegen gewisser Voraussetzungen ohne weitere Berechnung sofort festzulegen, so daß nicht für jeden Punkt der Linie eine Rechenschleife erforderlich ist. Die Zentrierung der einzusetzenden Konfigurationen erfolgt durch eine für den betreffenden Winkelbereich geeignete Vorbesetzung einer Entscheidungsschwelle für eine Stufe in der X- oder Y-Richtung. Die sich bei einer bevorzugten Aufteilung des Winkelbereichs zwischen 0° und 45° ergebende Konfigurationen sind in der nachstehenden Beschreibung eines anhand der Zeichnung verdeutlichten Ausführungsbeispiels näher erläutert. In der Zeichnung zeigen:

Fig. 1 eine Darstellung einer 45° -Linie mit dem erfindungsgemäßen Algorithmus und mit dem bekannten

Bresenham-Algorithmus,

Fig. 2 bis 4 Darstellungen für den Winkelbereich zwischen 38° und 45° , wobei Fig. 2 die obere Grenze dieses Bereiches, Fig. 3 einen mittleren Fall dieses Bereiches und Fig. 4 die untere Grenze dieses Bereiches zeigt.

Fig. 5 bis 7 Darstellungen gemäß Fig. 1 für den Winkelbereich von $18,4^\circ$ bis 38° , wobei Fig. 5 die obere Grenze, Fig. 6 einen mittleren Fall und Fig. 7 die untere Grenze dieses Winkelbereiches zeigen.

Fig. 8 bis 9 Darstellungen gemäß Fig. 1 für den Winkelbereich von $14,0^\circ$ bis $18,4^\circ$, wobei Fig. 8 die obere Grenze und Fig. 9 die untere Grenze dieses Winkelbereiches zeigen.

Fig. 10 bis 11 Darstellungen gemäß Fig. 1 für den Winkelbereich von $9,5^\circ$ bis $14,0^\circ$, wobei Fig. 10 die obere Grenze und Fig. 11 die untere Grenze dieses Winkelbereiches zeigen.

Fig. 12 bis 14 Darstellungen gemäß Fig. 1 für den Winkelbereich von $5,7^\circ$ bis $9,5^\circ$, wobei Fig. 12 die obere Grenze, Fig. 13 einen mittleren Fall und Fig. 14 die untere Grenze dieses Winkelbereiches zeigen.

Fig. 15 bis 17 Darstellungen gemäß Fig. 1 für den Winkelbereich von > 0 bis $5,7^\circ$, wobei Fig. 15 die obere Grenze, Fig. 16 einen mittleren Fall und Fig. 17 die untere Grenze dieses Bereiches zeigen.

In den Figuren bedeuten x normale Pixelpunkte mit voller Intensität, O Pixelpunkte mit einer ersten, verminderten Intensität (ca. zwei Drittel der vollen Intensität) und — Pixelpunkte mit einer zweiten, verminderten Intensität (etwa halbe Intensität).

In den nachstehenden Figuren sind jeweils die Darstellungen der Linien verschiedener Winkel oben nach dem erfindungsgemäßen Verfahren und unten im Vergleich dazu nach dem Bresenham-Algorithmus dargestellt.

Fig. 1 zeigt eine 45° -Linie, deren Darstellung nach dem Bresenham-Algorithmus an sich unproblematisch ist, weil die angesteuerten Pixelpunkte alle genau auf der darzustellenden Geraden liegen. Dennoch entsteht nach dem Bresenham-Algorithmus der Nachteil, daß die dargestellte Gerade eine geringere Helligkeit hat als eine horizontale Gerade, weil die Pixelpunkte einen um $\sqrt{2}$ größeren Abstand zueinander aufweisen als bei einer horizontalen Geraden. Nach dem erfindungsgemäßen Verfahren wird die 45° -Linie daher mit einem Anfangs- und Endpunkt mit normaler Intensität, im übrigen aber mit untereinander angeordneten Pixelpaaren dargestellt, die jeweils mit der ersten verminderten Intensität angesteuert sind. Dadurch wird gewährleistet, daß die resultierende Helligkeit der dargestellten Geraden der Helligkeit einer horizontalen Geraden entspricht.

Fig. 2 zeigt den Fall einer Geraden, die geringfügig unter 45° geneigt ist. Hier führt der Bresenham-Algorithmus zu einer deutlich erkennbaren Stufe. Fig. 3 und 4 zeigen, daß bei etwas größeren Abweichungen von der 45° -Richtung mehrere deutlich erkennbare Stufen auftreten. Bei dem erfindungsgemäßen Verfahren wird für diesen Winkelbereich ein Punkt mit normaler Intensität erzeugt, wenn dieser auf der Geraden liegt. Im Anschluß an den Punkt mit einer normalen Intensität wird in derselben Zeile ein Punkt mit der verminderten zweiten Intensität und in Neigungsrichtung darüber oder darunter ein Punkt mit der verminderten ersten Intensität gesetzt. Diese Konfiguration ist typisch für den Winkelbereich unter 45° bis hin zu etwa 38° , wie dies in Fig. 4 verdeutlicht ist. Zwischen diesen Konfigurationen befinden sich die aus Fig. 1 bekannten Pixelpaare mit der ersten verminderten Intensität. Für den nur wenig unter 45° liegenden, in Fig. 2 dargestellten Winkel, sind die Konfigurationen mit dem Punkt der normalen Intensität in Kombination mit dem Pixelpaar aus der ersten und der zweiten verminderten Intensität nur an den Endpunkten vorhanden, während im übrigen die Pixelpaare mit der ersten verminderten Intensität geschrieben werden, während an der unteren Grenze dieses Winkelbereiches nur noch jeweils ein Pixelpaar mit der ersten verminderten Intensität zwischen den für diesen Winkelbereich typischen Konfigurationen um die Punkte mit der normalen Intensität herum vorhanden sind. Aufgrund der aufgezeigten Gesetzmäßigkeit für diesen Winkelbereich ist es möglich, zusätzlich zu einem Pixelpunkt normaler Intensität sofort die benachbarten Pixelpaare mit den beiden verminderten Intensitäten sowie zumindest ein weiteres Pixelpaar mit der ersten verminderten Intensität zu setzen, ohne hierfür Rechenschritte ablaufen lassen zu müssen. Zur Zentrierung der Stufung wird die Entscheidungsschwelle für das Setzen eines neuen Punktes mit voller Intensität auf $(5 dy - 3 dx)$ vorbesetzt, während der Bresenham-Algorithmus die Entscheidungsschwelle $(2 dy - dx)$ vorbesetzt.

Die Fig. 5 bis 7 zeigen den Winkelbereich von $< 38^\circ \geq 18,4^\circ$. Verwendet werden hier nur Punkte mit normaler Intensität und die Pixelpaare mit der ersten verminderten Intensität. Mit abnehmendem Winkel wird die relative Anzahl der Pixelpaare mit der verminderten ersten Intensität im Vergleich zu den Punkten mit normaler Intensität vermindert. In Fig. 5 beträgt das Verhältnis 3 : 1, für den mittleren Fall dieses Winkelbereichs in Fig. 6 1 : 1 und für die untere Grenze, die in Fig. 7 dargestellt ist, 1 : 2. Die Fig. 8 und 9 zeigen einen Winkelbereich von $< 18,4^\circ$ bis $\geq 14,0^\circ$. Zwischen Punkten mit normaler Intensität befinden sich hier charakteristische Anordnungen eines Pixelpaares mit den beiden verminderten Intensitäten, daran anschließend ein Pixelpunkt mit der verminderten ersten Intensität.

Die Fig. 10 und 11 stellen den Winkelbereich $< 14,0^\circ \geq 9,5^\circ$ dar. Für diese kleineren Winkel ändert sich die charakteristische Konfiguration der Pixelanordnungen mit der verminderten Intensität durch zwei antiparallele Paare mit den beiden verminderten Intensitäten. Gegenüber dem in den Fig. 8 und 9 gezeigten Winkelbereich wird demzufolge ein Punkt mit verminderter zweiter Intensität hinzugefügt. Diese charakteristische Anordnung kann zur Verminderung der Rechenzeit ohne zusätzliche Rechenschritte gesetzt werden.

Fig. 12 bis 14 zeigen

Fälle aus dem flachen Winkelbereich von $< 9,5^\circ \geq 5,7^\circ$. Die von den Fig. 10 und 11 bekannte Anordnung der antiparallelen Pixelpaare mit den beiden verminderten Intensitäten ist für diesen Winkelbereich erweitert worden zu einer Konfiguration von zwei antiparallelen Pixelpaaren mit den beiden verminderten Intensitäten. Zwei parallelen Pixelpaaren dieser Art folgen somit zwei antiparallele Pixelpaare. Da anschließend immer wenigstens ein Normalpunkt folgt, kann dieser gleich mitgesetzt werden.

Die in den Fig. 15 bis 17 dargestellten Fälle noch geringerer Steigung verwenden eine Konfiguration aus drei parallelen Pixelpaaren mit den verminderten Intensitäten, denen sich drei antiparallele entsprechende Pixelpaare anschließen. Zu diesen Konfigurationen können benachbart gleich drei Normalpunkte mitgesetzt werden, da

— wie der Übergang von der Fig. 15 zu Fig. 16 und 17 zeigt, die Anzahl der Normalpunkte mit abnehmendem Winkel ansteigt, bis beim minimalen Winkel, der sich aus dem Höhenabstand $dy = 1$ für die Endpunkte über die gesamte Breite des Bildschirms ergibt, nur noch eine charakteristische Konfiguration mit den anschließenden Normalpunkten vorhanden ist.

5 Es ist ohne weiteres ersichtlich, daß für die anderen Winkelbereiche dieselben Gesichtspunkte gelten und die Geraden in analoger Weise aufgebaut werden können.

Überraschenderweise hat es sich gezeigt, daß der erfindungsgemäße Algorithmus trotz der zusätzlich gesetzten Pixelpunkte regelmäßig eine kürzere Rechenzeit benötigt als der Bresenham-Algorithmus. Nur für ungünstige Einzelfälle kommt es zu einer Rechenzeit, die geringfügig über der Rechenzeit für den Bresenham-Algorithmus liegt.

10 Ein kommentiertes Programm in der Programmiersprache C ist nachfolgend angegeben, aus dem sich die bildlichen Darstellungen der Fig. 1 bis 17 ergeben haben.

15

20

25

30

35

40

45

50

55

60

65

Programmiersprache C

Programmiersprache C

```

/*=====*/
/* Funktion zum Ziehen von Linien auf Rasterbildschirmen mit hoher Bild- */
/* Qualitaet durch 2-fach Antialiasing; bei guter Umsetzung erreicht diese */
/* Funktion die Schreibgeschwindigkeit der Bresenham-Routine. */
/* Alle Multiplikationen in diesem File lassen sich auf einfache Addi- */
/* tionen zurueckfuehren, sodass dieser Algorithmus nur mit Addition und */
/* Subtraktion und nur mit Integer-Werten auskommt! */
/* COPYRIGHT 1990 Klaus Bavendiek Bammelsburgerstr.4 1300 Braunschweig */
/*=====*/

void draw_line ( int xs, int ys, int xe, int ye)
{
    int dx, dy, /* Delta X bzw. Delta Y ( immer >= 0 ) */
    E, /* Entscheidungsschwelle, ob auch in Y-Richtung gezogen wird */
    x, y, /* aktueller Punkt, an dem ein Farbwert gesetzt wird */
    /*xs, ys, /* Start-Werte */
    /*xe, ye, /* End-Werte */
    d2y, /* um diesen Wert wird E erhoehrt, wenn nur in X-Richtung
    gezogen wird */
    d2y_d2x, /* um diesen Wert wird E erhoehrt, wenn in X- und
    Y-Richtung gezogen wird */
    dxx, dyy, /* Inkremente zum Erhoechen von X und Y (Rechner-abhaengig) */
    hxy, /* Hilfsvariable fuer dies und das */
    i; /* Schleifenzaehler in den for- und while-Schleifen

    FARBE Farbel, /* Variable vom Typ Farbe; die Normalfarbe zum Linienziehen */
    AliasFarbel, /* erste Antialiasing-Farbe ( ca. 0.72 * Intensitaet(Farbel) */
    AliasFarbe2; /* zweite Antialiasing-Farbe (ca. 0.48 * Intensitaet(Farbel) */
    /* Fuer die unterschiedlichen Farben muessen unterschied-
    /* liche Intensitaeten der Antialiasing-Farben 1&2 einge-
    /* stellt werden koennen, um ein optimales Bildergebnis zu
    /* erhalten. Es empfiehlt sich hier, entsprechenden Farben
    /* in der COLOR LOOKUP TABLE bereit zu halten.

/* =====*/

    moveto(xs, ys); /* zum Startwert begeben
    dx = xe - xs; /* Delta X ermitteln
    dy = ye - ys; /* Delta Y ermitteln

    if (abs(dx) >= abs(dy)) /*Betrag von Delta X ist groesser als Betrag von
    Delta Y. Der umgekehrte Fall laeuft aequivalent
    mit vertauschten X und Y Werten ab.

    {
        if ( dx < 0 ) /* Fuer negatives dx werden Anfangswerte und
        Endwerte von X und Y vertauscht.

        {
            dx = -dx;
            hxy = xs; xs = xe; xe = hxy;
            hxy = ys; ys = ye; ye = hxy;

        }

        dxx = 1; /* Stride in X-Richtung setzen; Rechner-abhaengig */
        dyy = 1; /* Stride in Y-Richtung setzen; Rechner-abhaengig */

        if ( dy == 0 ) /* genau 0 Grad; einfach gerade Linie zeichnen. */
        {
            Fall 0;
        }
        else
        {
            if ( dy < 0 ) /* DY-Werte fuer negative Steigung umsetzen */
            {
                dy = -dy; dyy=-dyy;
            }
            d2y = dy+dyy; /* um diesen Wert wird E erhoehrt, wenn nur in
            X-Richtung gezogen wird (else-Teil)

```

```

d2y_d2x = d2y-dx-dx; /* um diesen Wert wird E erhoeht, wenn in
                        X- und Y-Richtung gezogen wird (if-Teil) */
x = xs; y = ys;      /* Start-Werte setzen */

5  if ( d2y_d2x == 0 ) /* dx = dy, das entspricht genau 45 Grad */
    {
        Fall 1;
    }
    else
        /* != 45 Grad */
10  {
        hxy = d2y_d2x+d2y_d2x; /* Hilfsvariable auf 4dy-4dx setzen */
        if ( (dx+hxy) >=0 )
            /* (4dy > 3dx) =^ dy:dx >= 3:4 =^ 38 .. 45 Grad */
            {
                Fall 2;
            }
            else
                /* < 38 Grad */
15  {
                E = d2y+dy - dx; /* E ist zu Anfang 3dy-dx */
                if ( E >= 0 )
                    /* (3dy > dx) =^ dy:dx >= 1:3 =^ 18.4 .. 38 Grad */
                    {
20  {
                        Fall 3;
                    }
                    else
                        /* < 18.4 Grad */
                        {
25  {
                            E = E + dy; /* E ist hier 4dy-dx */
                            if ( E >=0 )
                                /* (4dy > dx) =^ dy:dx >= 1:4 =^ 14.0 .. 18.4 Grad */
                                {
                                    Fall 4;
                                }
                                else
30  {
                                    /* < 14.0 Grad */
                                    {
                                        hxy = d2y+d2y;
                                        d2y_d2x = d2y_d2x + hxy; /* s. Kommentar im Unterprg.*/
                                        if ( (d2y + E) >=0 )
                                            /* (6dy > dx) =^ dy:dx >= 1:6 =^ 9.5 .. 14.0 Grad */
                                            {
35  {
                                                Fall 5;
                                            }
                                            else
                                                /* < 9.5 Grad */
                                                {
40  {
                                                    d2y_d2x = d2y_d2x + hxy; /* siehe oben */
                                                    E = E + hxy; /* E ist jetzt 8dy-dx */
                                                    if ( (E+d2y) >= 0 ) /* 5.7 .. 9.5 Grad >= 1:10 */
                                                        /* (10dy > dx) =^ dy:dx >= 1:10 =^ 5.7..9.5 Grad */
                                                        {
45  {
                                                            Fall 6;
                                                        }
                                                        else
                                                            /* < 5.7 Grad < 1:10 */
                                                            {
50  {
                                                                d2y_d2x = d2y_d2x + hxy + hxy; /* siehe oben */
                                                                E = E + d2y; /* E ist jetzt 10dy-dx */
                                                                {
                                                                    Fall 7;
                                                                }
                                                                {
                                                                    /* else 5.7 .. 9.5 Grad */
                                                                    /* else von 9.5 .. 14.0 Grad */
                                                                    /* else von 14.0 .. 18.4 Grad */
                                                                    /* else von 18.4 .. 38 Grad */
                                                                    /* else von 38 .. 45 Grad */
                                                                    /* else von genau 45 Grad */
                                                                    /* else von genau 0 Grad */
                                                                    /* ===== */
55  /* =====
60  /* if (dx >= dy) */

```

```

/* ===== 0 .. 45 Grad Linien ===== */
else
/* ===== 45 .. 90 Grad Linien ===== */
{
    /* Diese Linien verlaufen analog zu den 0 .. 45 Grad-Linien; es sind
    jedoch die X- mit den entsprechenden Y-Werten zu tauschen. */
    /* else (dx >= dy) */
    /* end of draw_line (...) */
}

/* ===== */
/* Nun folgt die Auflistung der einzelnen Faelle: */
/* ===== */

Fall 0; /* genau 0 Grad */
{
    for ( x=xs;x<xe;x=x+dx ) /* von xs bis xe Punkte in Reihe setzen */
    {
        put_pixel (x,y,Farbe1); /* Setzen eines Punktes an Pos. X-Y */
        /* mit der Normal-Farbe */
        /* genau 0 Grad */
    }
}

Fall 1; /* genau 45 Grad */
{
    put_pixel (x,y,Farbe1); /* Anfangspunkt mit Normalfarbe setzen */
    y = y + dyy; /* eine Zeile nach unten springen */
    put_pixel (x,y,AliasFarbe1); /* Unter dem Anfangspunkt Punkt mit
    /* der 1. Alias-Farbe setzen */
    for ( x=xs+dx; x<xe; x=x+dx )
    {
        /* vom 2. Punkt bis vorletzten Punkt */
        put_pixel (x,y,AliasFarbe1); /* immer 2 Punkte mit der 1. Alias- */
        y = y + dyy; /* Farbe untereinander setzen. Die */
        put_pixel (x,y,AliasFarbe1); /* Gesamt-Helligkeit entspricht ca. */
    } /* for */ /* einer 0-Grad-Linie mit Farbe1. */

    put_pixel (x,y,Farbe1); /* Endpunkt mit Normal-Farbe setzen */
    /* genau 45 Grad */
}

Fall 2; /* Fall 2 von 38 - 45 Grad */
/* Die Linien um 45 Grad erfordern besonderen Aufwand, da trotz Setzen
/* von AliasPunkten Treppenstufen deutlich erkennbar sind. Hier bietet nur
/* die 2. Aliasfarbe Abhilfe. Es werden in diesem Winkelbereich immer neben
/* einem Normal-Punkt links und rechts 2 Punkte mit der 2. Alias-Farbe ge-
/* setzt. Aufgrund des Winkels kann vorausgesagt werden, dass hinter jedem
/* Normal-Punkt mindestens 2*2 Punkte mit 1. Alias-Farbe gesetzt werden.
/* Das wird bei dem "else-Teil" der if-Abfrage genutzt und diese Punkte
/* gleich mitgesetzt. Entsprechend muss das d2y, das hier zum E addiert
/* wird, um 2* den Anteil der sonst zu durchlaufenden if-Schleife erhoeht
/* werden. Die Entscheidungsschwelle E (im Referenzfall: 3dy-dx) muss auf
/* 5dy-3dx vorbesetzt werden, damit die Stufung zentriert wird. Der if-
/* Teil der for-Schleife entspricht genau der 45-Grad Variante, d.h. es
/* werden jeweils 2 Punkte mit der 1. AliasFarbe nebeneinander gesetzt
/* (s. Fall 1).
{
    E = hxy + dy + dx;
    d2y = d2y + d2y_d2x + hxy;
    put_pixel (x,y,Farbe1);
    x = x + dx;
    put_pixel (x,y,AliasFarbe2);
    for ( i=2; i<dx; i++ )
    {
        if (E>0)
        {
            y = y + dyy;
            put_pixel (x,y,AliasFarbe1);
            x = x + dx;
            put_pixel (x,y,AliasFarbe1);
        }
    }
}

```

```

      E = E + d2y_d2x;
    }
    else
    {
5       y = y + dyy;
        put_pixel (x,y, AliasFarbe2);
        x = x + dxx;
        put_pixel (x,y, Farbe1);
        x = x + dxx;
10       put_pixel (x,y, AliasFarbe2);
        y = y + dyy;
        put_pixel (x,y, AliasFarbe1);
        x = x + dxx;
        put_pixel (x,y, AliasFarbe1);
        y = y + dyy;
15       put_pixel (x,y, AliasFarbe1);
        x = x + dxx;
        put_pixel (x,y, AliasFarbe1);
        E = E + d2y;
        i = i + 3; /* es wurden 4 Punkte in x-Richtung in einem */
                    /* Durchlauf gesetzt. Der Schleifenzaehler i */
20      } /* for */
        put_pixel ( xe-dxx,y, AliasFarbe2);
        put_pixel ( xe,y, Farbe1);
    } /* Fall 2 von 38 - 45 Grad */

25 Fall 3; /* Fall 3 von 18.4 - 38 Grad */
/* Dieses sind die Winkel, die den wenigsten Aufwand erfordern. Sie stel- */
/* len ca. die Haelfte aller Faelle dar. Es wird entweder ein Punkt in der */
/* Normal-Farbe ( if-Teil ) oder 2 Punkte mit der 1. Alias-Farbe unterein- */
/* ander ( else-Teil ) gesetzt. Die Entscheidungsschwelle wird im Gegen- */
30 /* satz zum Bresenham (E=2dy-dx) auf E=3dy-dx gesetzt; der zu E zu addie- */
/* rende Term in dem if-Teil der for-Schleife (d2y_d2x) ist (wie der Name */
/* sagt) d2y-d2x; in dem else-Teil ist dieser Term (d2y) 2dy. Alle anderen */
/* Faelle leiten sich von diesem Fall ab; um die Anzahl der Variablen ge- */
/* ring zu halten, wurden d2y_d2x (immer fuer den if-Teil) und d2y (immer */
35 /* fuer den else-Teil) beibehalten und jeweils den veraenderten Winkeln */
/* durch Addition von entsprechenden Anteilen von dx und dy angepasst. */
    put_pixel (x,y, Farbe1);
    for ( i=1; i<dx; i++ )
    {
40         if (E>0)
        {
            x = x + dxx;
            put_pixel (x,y, AliasFarbe1);
            y = y + dyy;
45         put_pixel (x,y, AliasFarbe1);
            E = E + d2y_d2x;
        }
        else
        {
50         x = x + dxx;
            put_pixel (x,y, Farbe1);
            E = E + d2y;
        }
    } /* for */
    put_pixel ( xe,ye, Farbe1);
55 } /* Fall 3 von 18.4 - 38 Grad */

Fall 4; /* Fall 4 von 14.0 - 18.4 Grad */
/* In diesem Winkelbereich wuerden im Fall 3 zwischen je 2 Alias-Punkten */
/* mindestens noch 2 Normal-Punkte liegen. Deshalb kann, um die Linienqua- */
/* litaet weiter zu verbessern, in diesem Fall der untere Aliaspunkt mit */
60 /* der 1. Aliasfarbe durch einen Aliaspunkt mit der 2. Aliasfarbe ersetzt */

```



```

/* und neben diesen Punkt auch noch ein Aliaspunkt mit der 1. Aliasfarbe */
/* gesetzt werden. Da hier 2 Punkte in X-Richtung hintereinander gesetzt */
/* werden, muss d2y_d2x im if-Teil um ein d2y aus dem else-Teil erhoeht */
/* werden, da dieser Punkt im gleichen Durchlauf mit gesetzt wird. Der */
/* Schleifenzaehler i muss daher im if-Teil auch um 1 erhoeht werden. */
5
{
    d2y_d2x = d2y_d2x + d2y;
    put_pixel (x,y,Farbel);
    for ( i=1; i<dx; i++ )
    {
        if (E>0)
        10
        {
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            y = y + dyy;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            E = E + d2y_d2x;
            i = i + 1;
        }
        else
        20
        {
            x = x + dxx;
            put_pixel (x,y,Farbel);
            E = E + d2y;
        }
    }
    /* for */
    put_pixel ( xe, ye, Farbel);
    /* Fall 4 von 14.0 - 18.4 Grad */

Fall 5; /* Fall 5 von 9.5 .. 14.0 Grad */
/* Entspricht vollstaendig dem Fall 4, traegt aber zusaetzlich der Linien- */
/* intensitaet aufgrund des Winkel Rechnung, in dem zusaetzlich zum Fall 4 */
/* neben dem oberen Aliaspunkt ein weiterer Aliaspunkt mit der 2. Alias- */
/* farbe gesetzt wird. Fuer diesen und fuer alle weiteren Faelle gilt, */
/* dass ueber- oder unterhalb eines jeden Punktes mit der 1. Aliasfarbe */
/* ein Punkt mit der 2. Aliasfarbe gesetzt wird. Hier werden also die 2 */
/* Punkte mit Normalfarbe um einen Bresenham-Sprung gegen 2*2 Aliaspunkte */
/* getauscht. Um bereits einen Punkt vor dem Bresenham-Sprung den ersten */
/* Aliaspunkt setzen zu koennen, muss das E um genau einen Schritt in x- */
/* Richtung erweitert werden ( im uebergeordneten Programm schon erfolgt). */
30
{
    put_pixel (x,y,Farbel);
    for ( i=2; i<dx; i++ )
    {
        if (E>0)
        {
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,AliasFarbe2);
            y = y + dyy;
            x = x - dxx;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,Farbel);
            E = E + d2y_d2x;
            i = i + 2;
        }
        else
        {
            x = x + dxx;
            put_pixel (x,y,Farbel);
        }
    }
}

```

```

      E = E + d2y;
    }
    /* for */
    put_pixel ( x,y, Farbel);
5      /* Fall 5 von 9.5 .. 14.0 Grad */

Fall 6:
/* Fall 6 von 5.7 .. 9.5 Grad */
/* Um bei derart flachen Winkeln eine Stufe zu vertuschen, genuegen hier 2 */
/* Aliaspunkte-Paerchen nebeneinander wie im Fall 5 nicht mehr. Deshalb */
/* werden 4 Aliaspunkte-Paerchen nebeneinander gesetzt. Im uebergeordneten */
10 /* Programm wurde das d2y_d2x schon um die zusaetzlichen beiden d2y und */
/* ebenfalls das E um diesen Wert erhoehrt. Da bei diesen flachen Winkeln */
/* zwischen den Alias-Punkten noch mindestens 2 Normal-Punkte gesetzt wer- */
/* den, wird ( Vorsicht am Anfang und Ende der Linie ) ein Normal-Punkt */
/* in einem Durchlauf gleich mitgesetzt. (Rechenzeitvorteil). Auch hier */
15 /* muss der Schleifenzaehler entsprechend der Punkteanzahl erhoehrt werden. */
    put_pixel (x,y, Farbel);
    x = x + dxx;
    hxy = -dxx-dxx-dxx;
    put_pixel (x,y, Farbel);
20    for ( i=2; i<dx; i++ )
    {
        if (E>0)
        {
            x = x + dxx;
            put_pixel (x,y, AliasFarbel);
25            x = x + dxx;
            put_pixel (x,y, AliasFarbel);
            x = x + dxx;
            put_pixel (x,y, AliasFarbe2);
            x = x + dxx;
30            put_pixel (x,y, AliasFarbe2);
            y = y + dyy;
            x = x + hxy;
            put_pixel (x,y, AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y, AliasFarbe2);
35            x = x + dxx;
            put_pixel (x,y, AliasFarbel);
            x = x + dxx;
            put_pixel (x,y, AliasFarbel);
            x = x + dxx;
            put_pixel (x,y, Farbel);
40            E = E + d2y_d2x;
            i = i + 4;
        }
        else
        {
45            x = x + dxx;
            put_pixel (x,y, Farbel);
            E = E + d2y;
        }
    }
    /* for */
50    put_pixel ( x,y, Farbel);
    /* Fall 6 von 5.7 .. 9.5 Grad */

Fall 7:
/* Fall 7 von 0 .. 5.7 Grad */
/* Bei diesen Winkeln ist der meiste Aufwand erforderlich, um die Stufen */
/* zu verbergen. Zusaetzlich zum Fall 6 werden 2 weitere Alias-Paerchen */
55 /* gesetzt, sodass insgesamt 6 Alias-Paerchen nebeneinander stehen. Da je- */
/* dem Alias-Paerchen mindestens 4 Normal-Punkte folgen, werden 3 Normal- */
/* Punkte in einem Durchlauf gleich mitgesetzt. (Alle 4 Punkte mitzusetzen */
/* wuerden in unguenstigen Faellen zu einem Ueberschreiten der Endpunkte */
/* fuehren.) Die Additionswerte zu E wurden bereits im uebergeordneten */
/* Programm gesetzt. Der Schleifenzaehler wird jeweils um 8 zusaetzliche */
60

```

```

/* Punkte erhoeht.
{
    put_pixel (x,y,Farbel);
    x = x + dxx;
    hxy = -dxx-dxx-dxx-dxx-dxx;
    put_pixel (x,y,Farbel);
    for ( i=3; i<dx; i++ )
    {
        if (E>0)
        {
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbe2);
            y = y + dy;
            x = x + hxy;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbe2);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,AliasFarbel);
            x = x + dxx;
            put_pixel (x,y,Farbel);
            x = x + dxx;
            put_pixel (x,y,Farbel);
            x = x + dxx;
            put_pixel (x,y,Farbel);
            E = E + d2y_d2x;
            i = i + 8;
        }
        else
        {
            x = x + dxx;
            put_pixel (x,y,Farbel);
            E = E + d2y;
        }
    } /* for */
    put_pixel ( xe-dxx,y, Farbel);
    put_pixel ( xe,y, Farbel);
} /* Fall 7 von 0 .. 5.7 Grad */
/...../

```

Patentansprüche

1. Verfahren zur grafischen Darstellung einer zwischen einem Anfangspunkt und einem Endpunkt verlaufenden Geraden auf einem Bildschirm unter Verwendung eines die Pixel des Bildschirms ansteuernden Algorithmus, **gekennzeichnet durch** folgende Verfahrensschritte
 - Ermittlung des Abstandes des Anfangspunktes vom Endpunkt in X- und Y-Richtung
 - Unterscheidung verschiedener Winkelbereiche
 - für Winkel > 0° gegenüber der Horizontalen Ersetzen von wenigstens einem normal angesteuerten Pixelpunkt durch eine Mehrzahl von Pixelpunkten mit einer verringerten Intensität in für die unterschiedenen Winkelbereiche unterschiedlichen Konfigurationen.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Ersetzung der normal angesteuerten Pixelpunkte mit Konfigurationen von Pixelpunkten mit verringerter Intensität unter dem Gesichtspunkt der gleichen Helligkeit für unter verschiedenen Winkeln dargestellten Geraden erfolgt.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß die verringerte Intensität 65% – 80% der normalen Intensität beträgt.
4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß wenigstens zwei unterschied-

lich verringerte Intensitäten vorgesehen werden.

5. Verfahren nach Anspruch 3 und 4, dadurch gekennzeichnet, daß die zweite verringerte Intensität 45% – 60% der normalen Intensität ausmacht.

6. Verfahren nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß eine Unterscheidung in mindestens drei, vorzugsweise fünf Winkelbereiche zwischen 0° und 45° vorgenommen wird.

7. Verfahren nach Anspruch 6, dadurch gekennzeichnet, daß die Winkel 0° und 45° gesondert behandelt werden.

8. Verfahren nach einem der Ansprüche 1 bis 7, dadurch gekennzeichnet, daß in einem Winkelbereich zwischen 0° und 5,7° im Bereich einer sonst auftretenden Pixelstufe sechs Pixelpunkte durch paarweise untereinander angeordnete Pixelpunkte mit jeweils der ersten und der zweiten verminderten Intensität gesetzt werden.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, daß im Anschluß an die paarweisen Punkte vermindelter Intensität drei Pixelpunkte normaler Intensität sofort mitgesetzt werden.

10. Verfahren nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, daß in einem Winkelbereich zwischen 5,7° und 9,5° im Bereich einer sonst auftretenden Pixelstufe vier Pixelpunkte durch paarweise untereinander angeordnete Pixelpunkte mit jeweils der ersten und zweiten verminderten Intensität gesetzt werden.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, daß im Anschluß an die paarweisen Punkte vermindelter Intensität zwei Pixelpunkte normaler Intensität sofort mitgesetzt werden.

12. Verfahren nach einem der Ansprüche 1 bis 11, dadurch gekennzeichnet, daß im Winkelbereich zwischen 9,5° und 14° im Bereich einer sonst auftretenden Pixelstufe zwei Pixelpunkte durch paarweise untereinander angeordnete Pixelpunkte mit jeweils der ersten und der zweiten verminderten Intensität gesetzt werden.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, daß im Anschluß an die paarweisen Punkte vermindelter Intensität ein Pixelpunkt normaler Intensität sofort mitgesetzt wird.

14. Verfahren nach einem der Ansprüche 1 bis 13, dadurch gekennzeichnet, daß im Winkelbereich zwischen 14° und 18,4° im Bereich einer sonst auftretenden Stufe zwei Pixelpunkte durch eine Konfiguration mit einem Punktepaar der beiden verschiedenen verringerten Intensitäten und einem Punkt mit der verringerten ersten Intensität ersetzt werden.

15. Verfahren nach Anspruch 14, dadurch gekennzeichnet, daß neben der Konfiguration der Punkte mit den verringerten Intensitäten sofort ein Pixelpunkt normaler Intensität mitgesetzt wird.

16. Verfahren nach einem der Ansprüche 1 bis 15, dadurch gekennzeichnet, daß im Winkelbereich von 18,4° bis 38° ein Pixelpunkt im Bereich einer sonst auftretenden Stufe durch ein Paar untereinander angeordneter Pixelpunkte mit der verminderten ersten Intensität ersetzt wird.

17. Verfahren nach einem der Ansprüche 1 bis 16, dadurch gekennzeichnet, daß im Winkelbereich von 38° bis 45° neben normalen Pixelpunkten ein Paar übereinander angeordneter Pixelpunkte mit den beiden verminderten Intensitäten und wenigstens ein Paar übereinander angeordneter Pixelpunkte mit der ersten verminderten Intensität angeordnet sind.

18. Verfahren nach einem der Ansprüche 1 bis 17, dadurch gekennzeichnet, daß bei dem Winkel 45° alle Pixelpunkte außer den Endpunkten durch Paare untereinander angeordneter Pixelpunkte mit der verminderten ersten Intensität ersetzt sind.

19. Verfahren nach einem der Ansprüche 1 bis 18, dadurch gekennzeichnet, daß der Algorithmus ohne Multiplikations- und Divisionsschritte ausgebildet wird.

20. Verfahren nach Anspruch 19, dadurch gekennzeichnet, daß die Unterteilung in die verschiedenen Winkelbereiche durch Additionen und Subtraktionen der Abstände von Anfangs- und Endpunkt in X- und Y-Richtung bzw. hieraus durch Addition und/oder Subtraktion gebildeter Hilfsgrößen erfolgt.

21. Verfahren nach einem der Ansprüche 1 bis 20, dadurch gekennzeichnet, daß die Verringerung der Intensität in Abhängigkeit von der jeweiligen Farbe nach Farbtabelle gewählt wird.

22. Verfahren nach einem der Ansprüche 1 bis 21, dadurch gekennzeichnet, daß durch eine geeignete Vorbesetzung einer Entscheidungsschwelle (E) für eine Stufe in X- oder Y-Richtung eine Zentrierung für die eingesetzte Konfiguration aus den Punkten mit der verminderten Intensität vorgenommen wird.

Hierzu 17 Seite(n) Zeichnungen

— Leerseite —

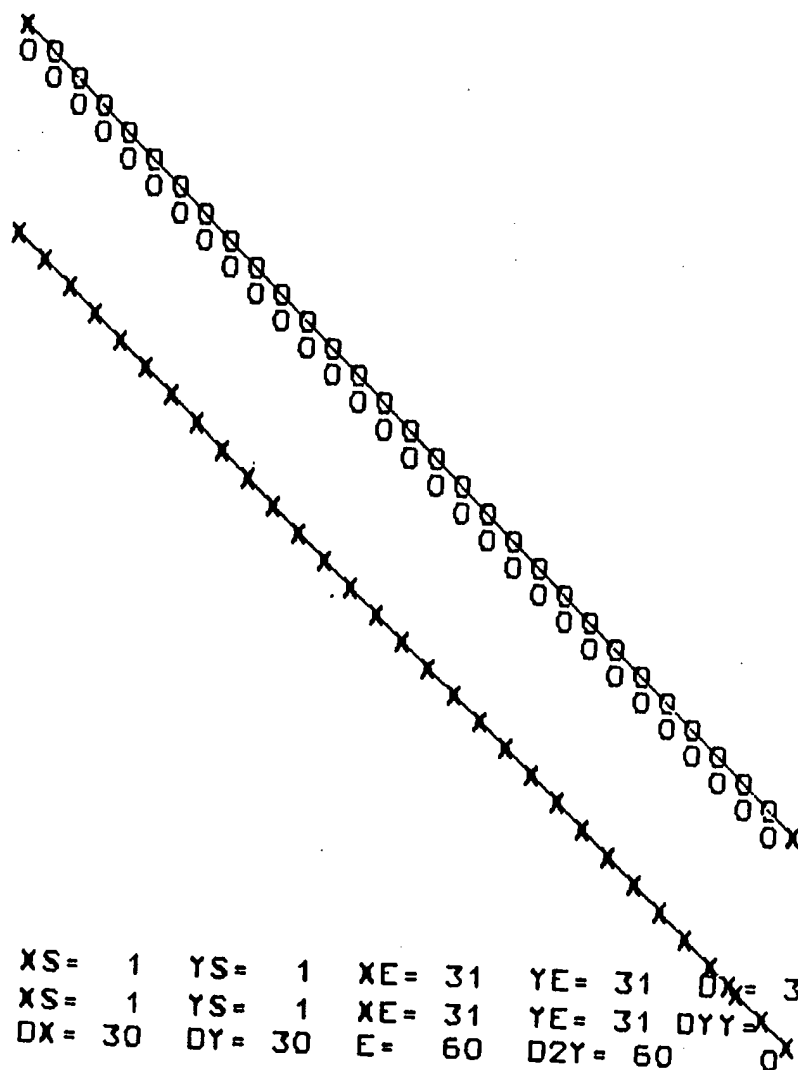


Fig 1

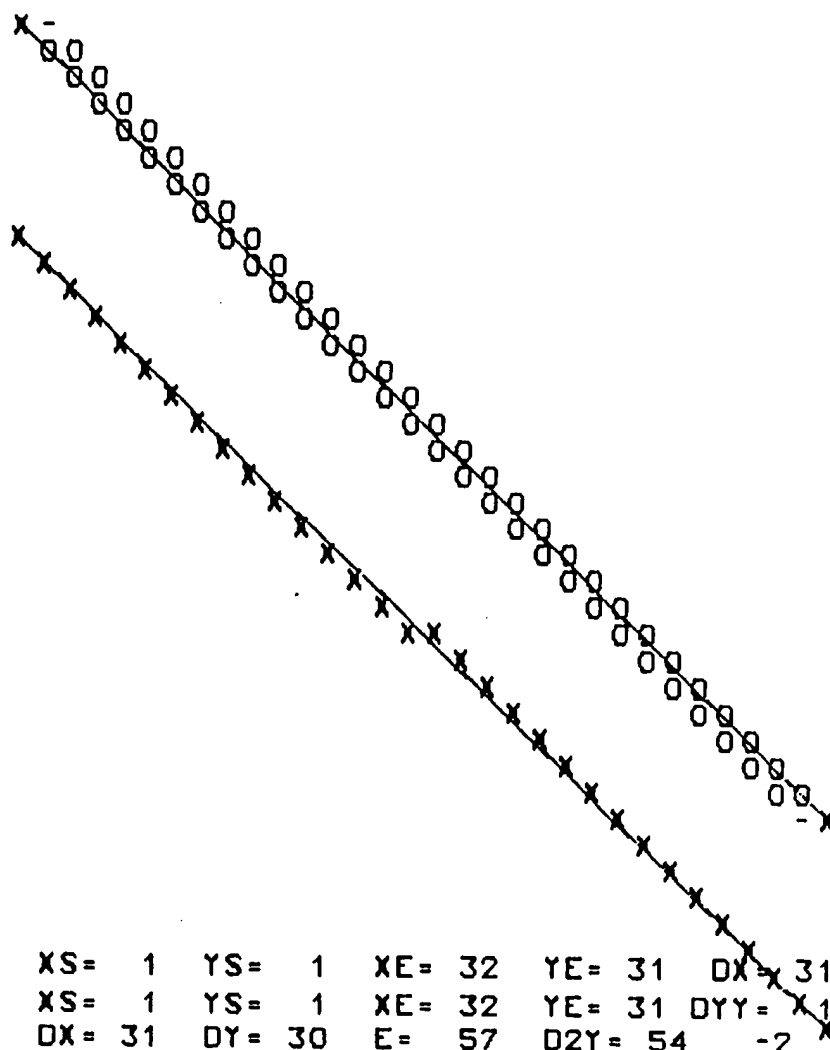
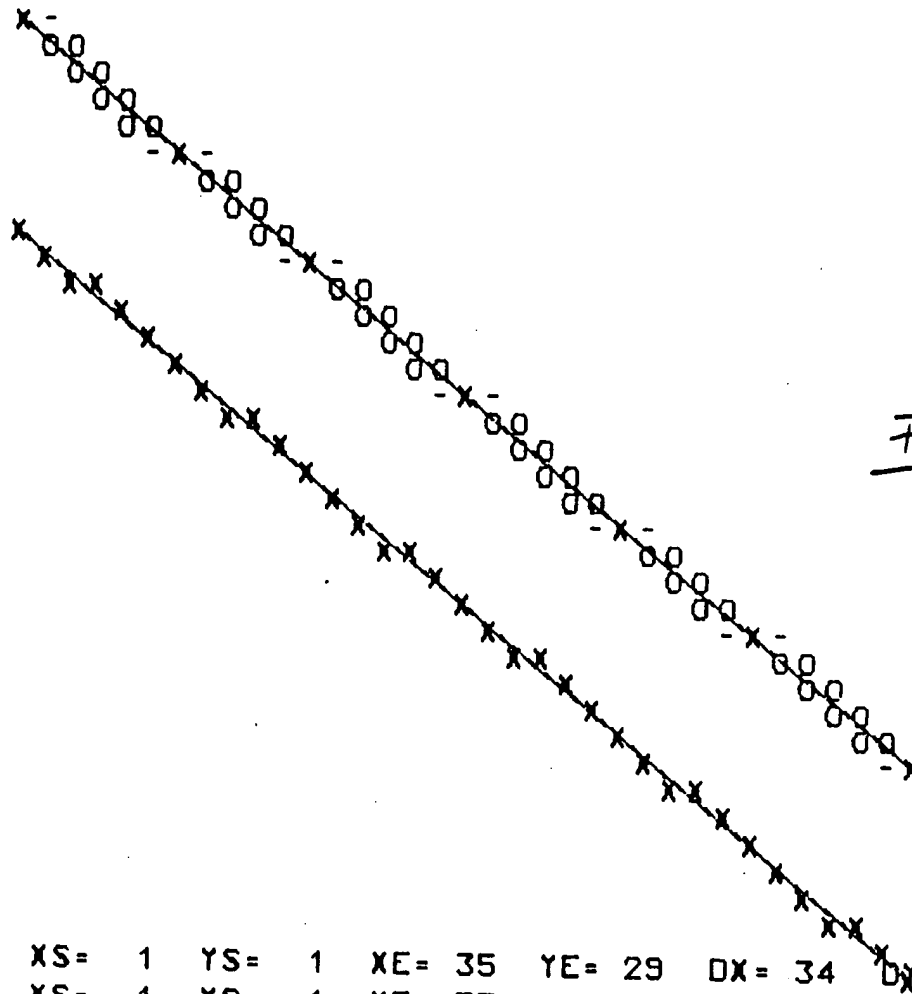
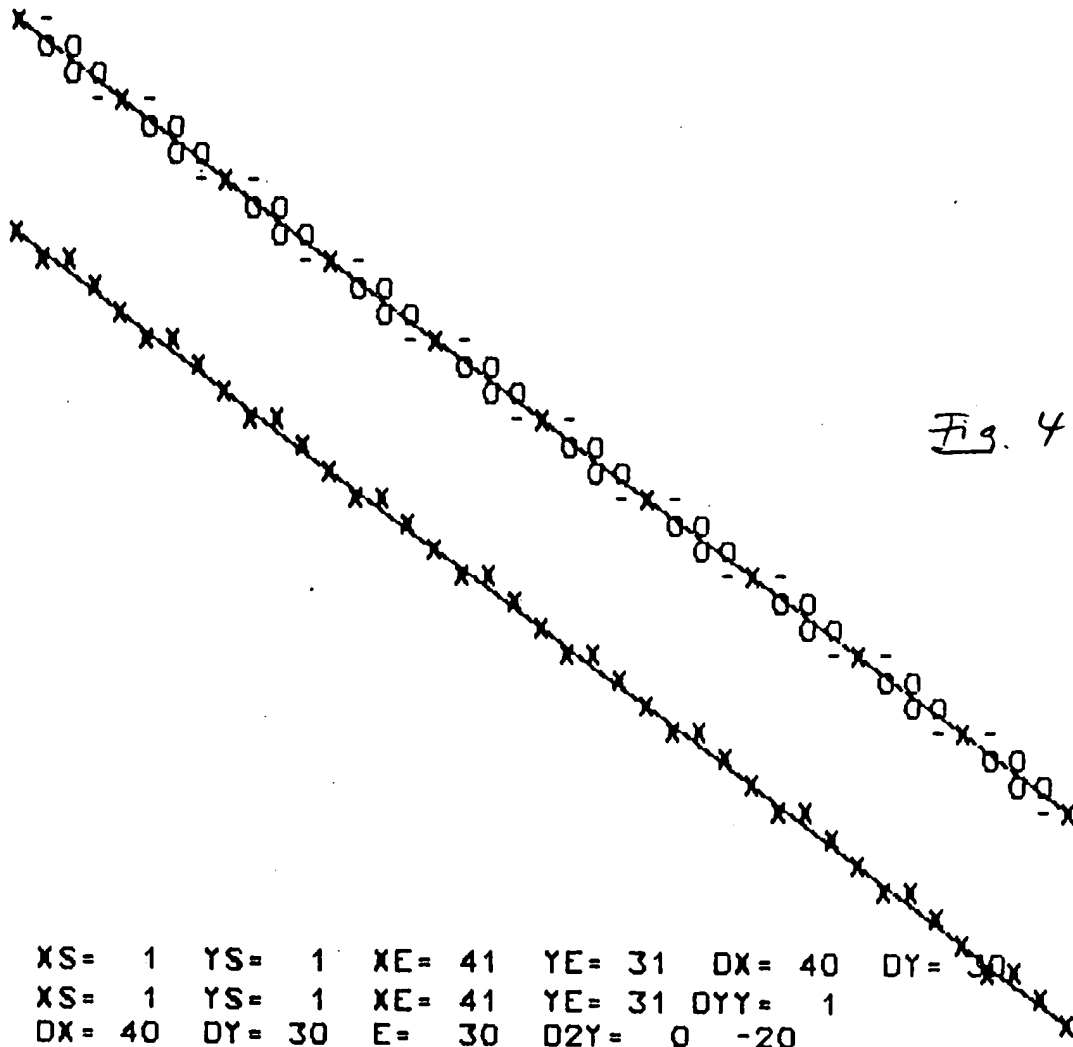
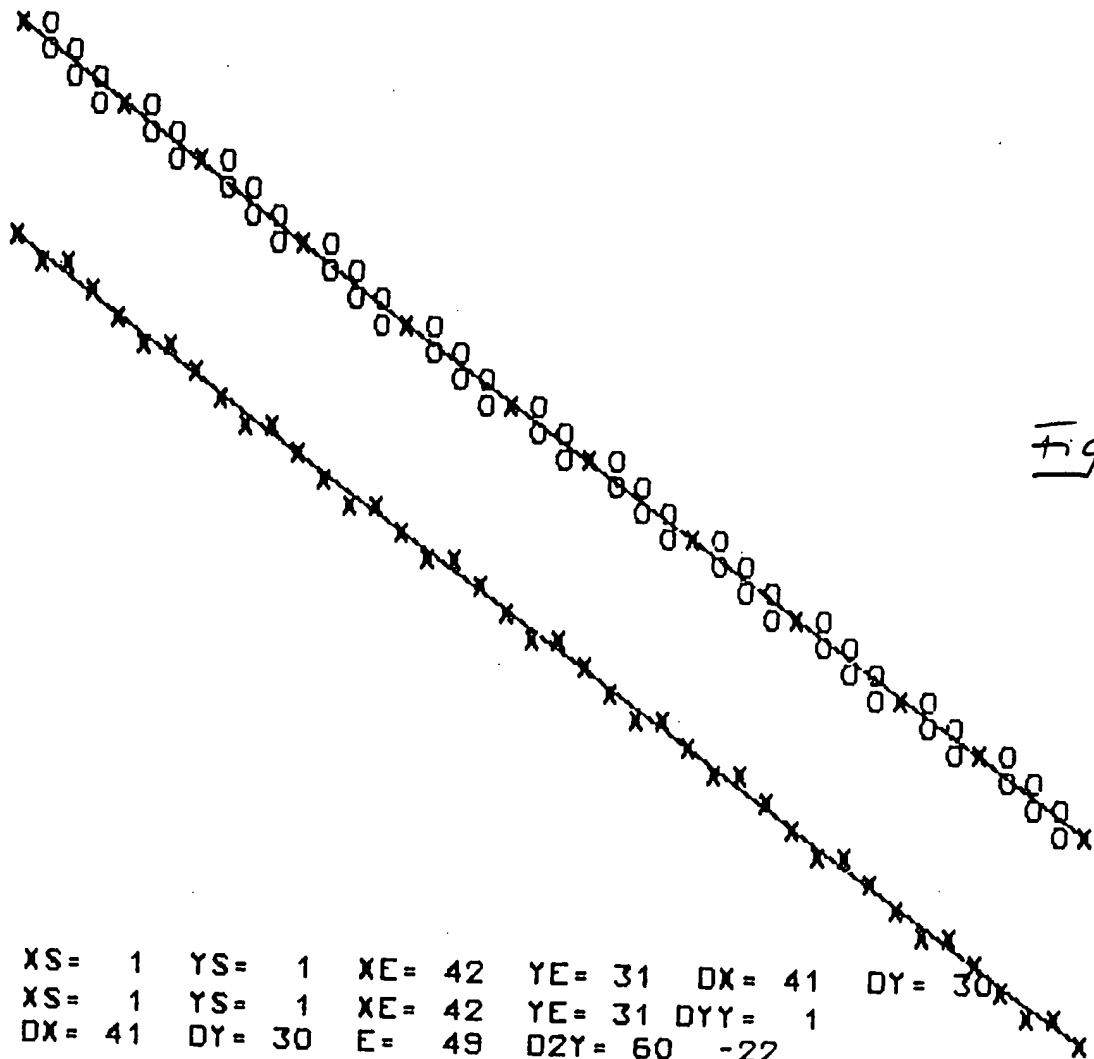


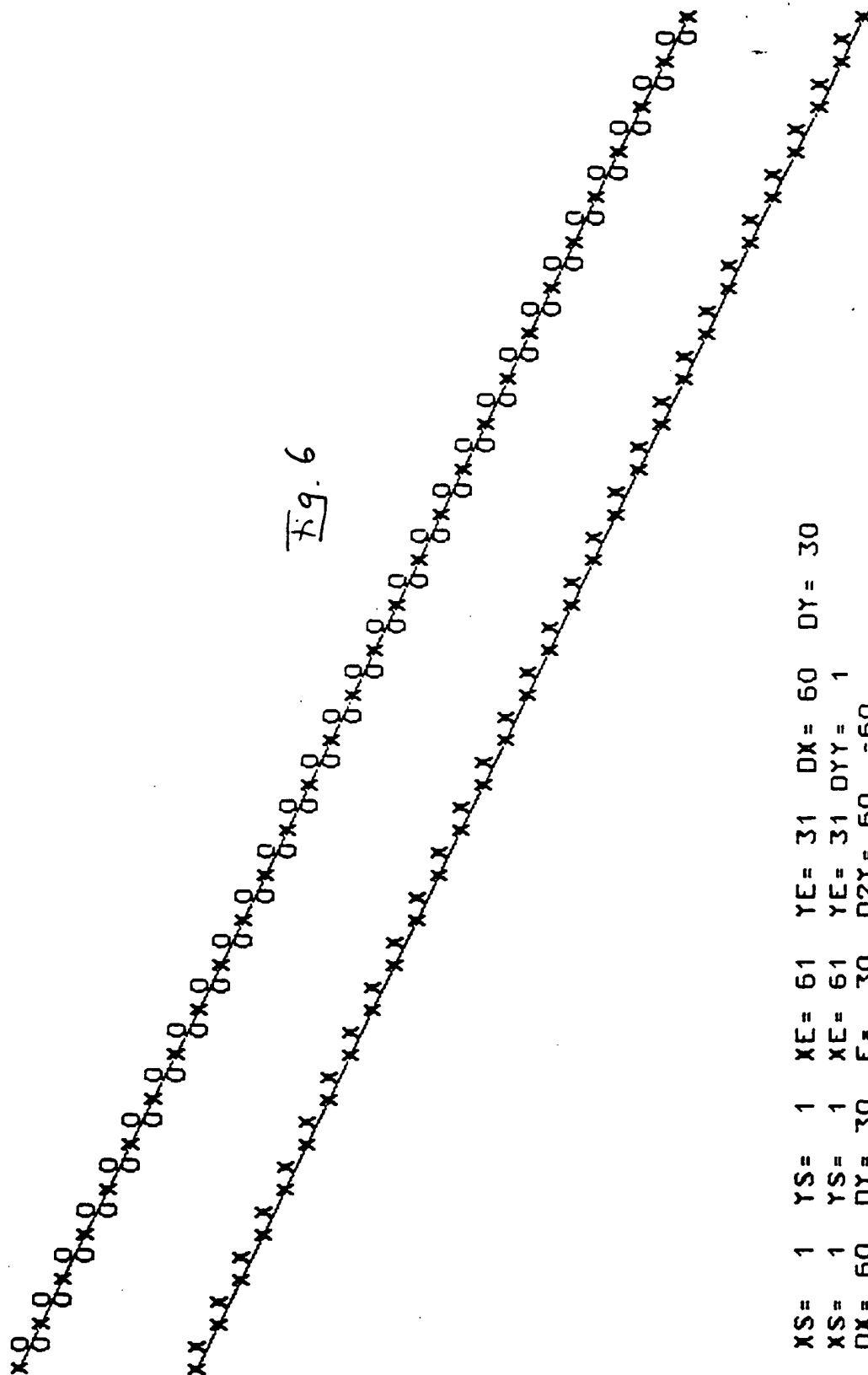
Fig. 2



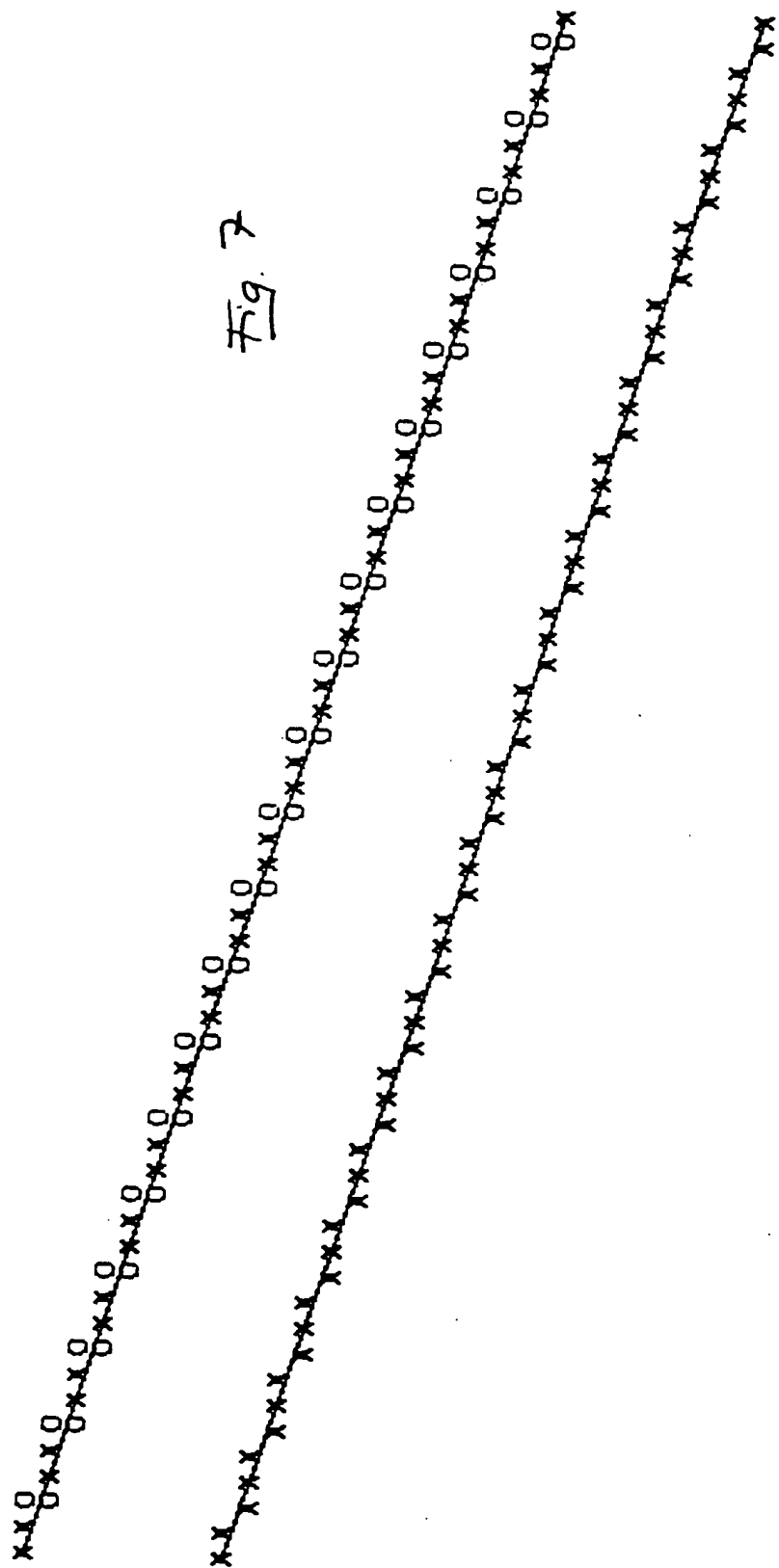
XS= 1 YS= 1 XE= 35 YE= 29 DX= 34 DY= 28
XS= 1 YS= 1 XE= 35 YE= 29 DYY= 1
DX= 34 DY= 28 E= 38 D2Y= 20 -12



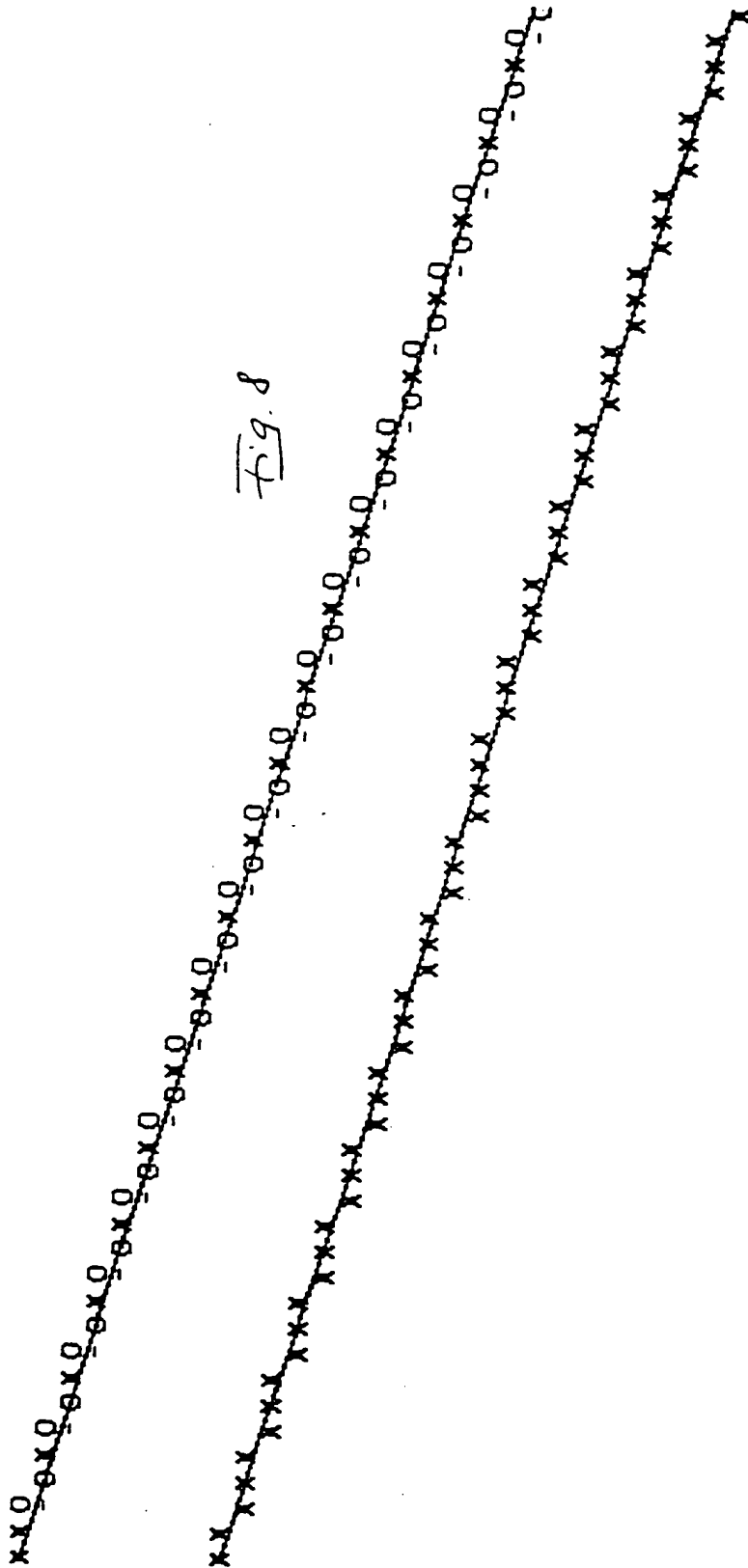




XS= 1 YS= 1 XE= 61 YE= 31 DX= 60 DY= 30
 XS= 1 YS= 1 XE= 61 YE= 31 DYY= 1
 DX= 60 DY= 30 E= 30 DZY= 60 -60



XS= 1 YS= 1 XE= 61 YE= 21 DX= 60 DY= 20
XS= 1 YS= 1 XE= 61 YE= 21 DYY= 1
DX= 60 DY= 20 E= 0 D2Y= 40 -80



XS= 1	YS= 1	XE= 62	YE= 21	DX= 61	DY= 20
XS= 1	YS= 1	XE= 62	YE= 21	DYY= 1	
DX= 61	DY= 20	E= -1	DZY= 40	-42	

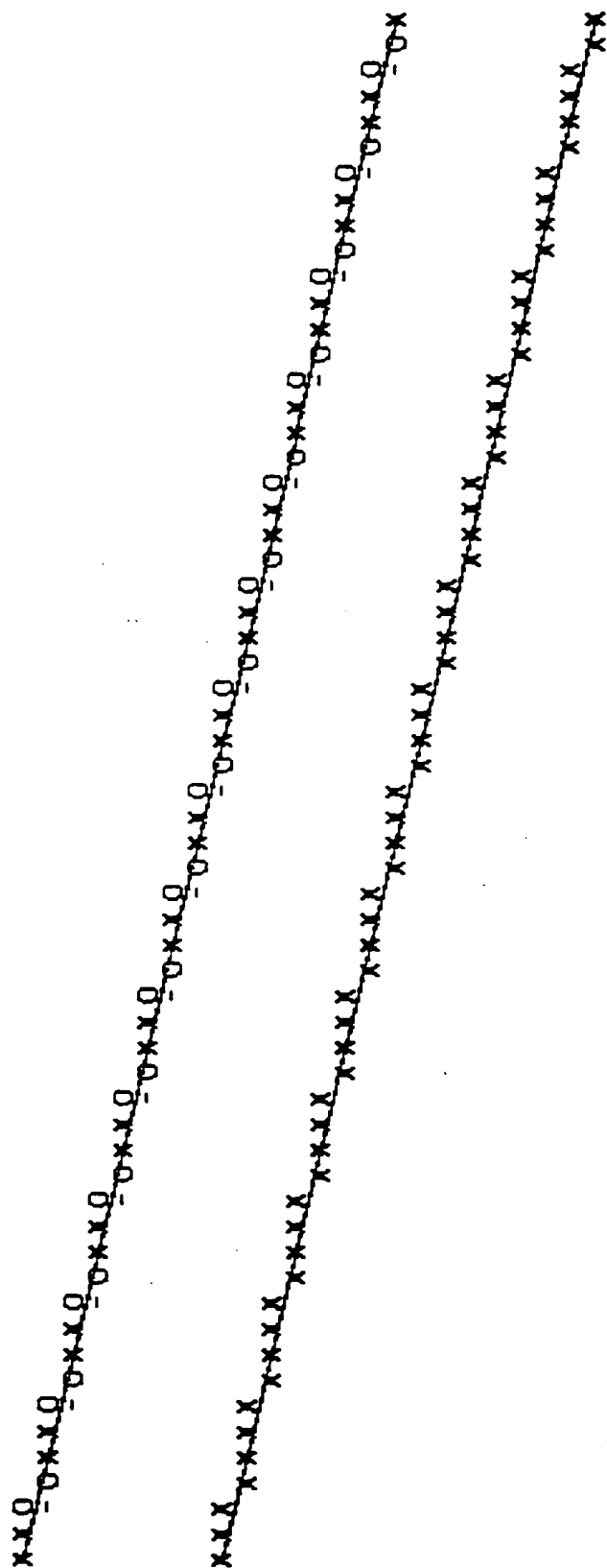


Fig. 9

XS= 1 YS= 1 XE= 61 YE= 16 DX= 60 DY= 15
XS= 1 YS= 1 XE= 61 YE= 16 DYY= 1
DX= 60 DY= 15 E= -15 D2Y= 30 -60

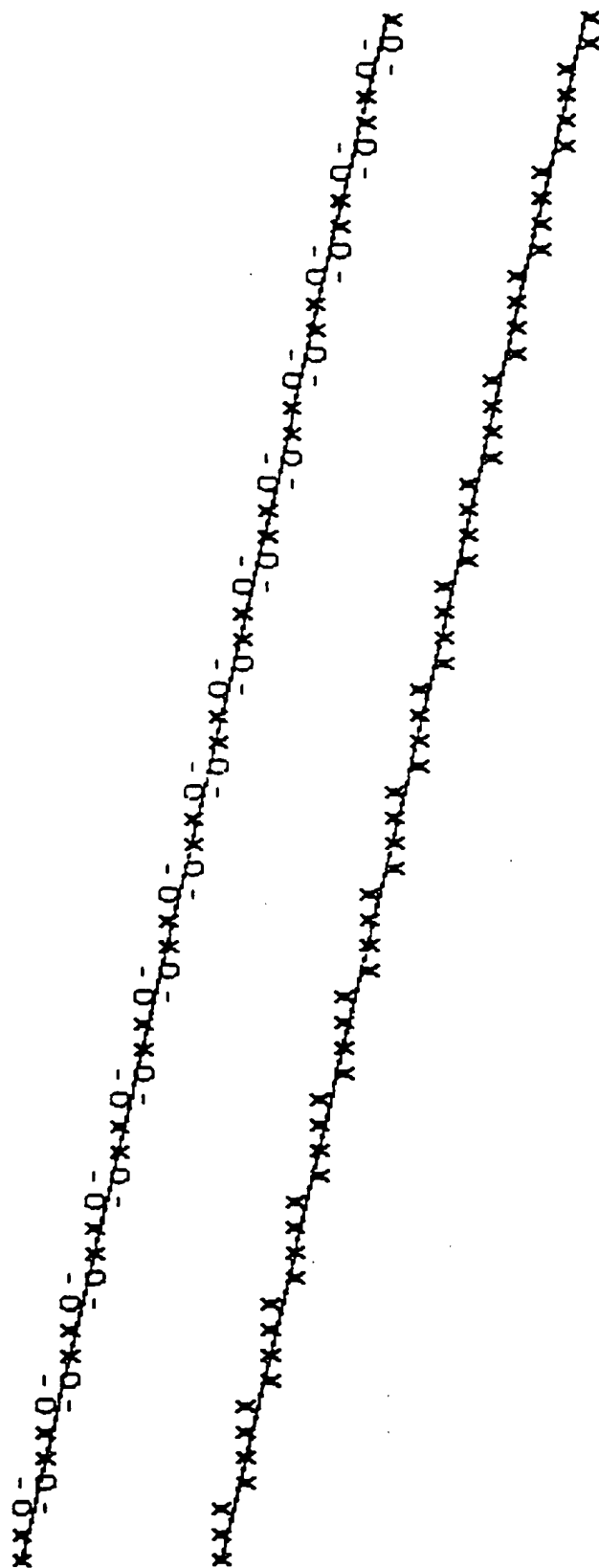


Fig. 10

XS= 1 YS= 1 XE= 62 YE= 16 DX= 61 DY= 15
 XS= 1 YS= 1 XE= 62 YE= 16 DYY= 1
 DX= 61 DY= 15 E= -1 D2Y= 30 -62

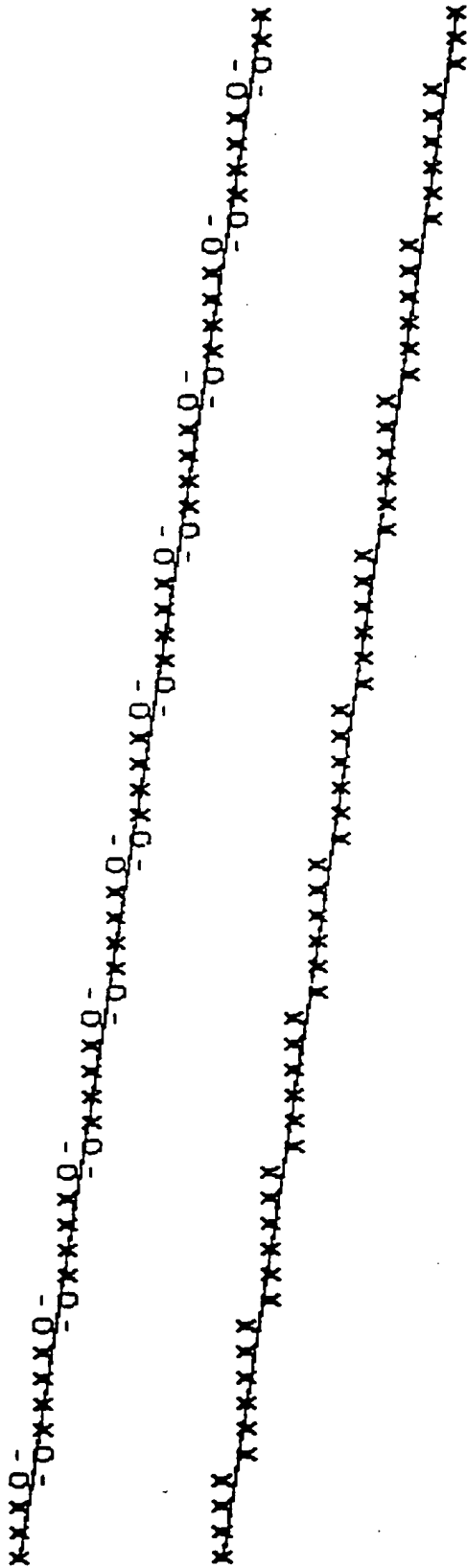


Fig. 11

XS= 1 YS= 1 XE= 61 YE= 11 DX= 60 DY= 10
 XS= 1 YS= 1 XE= 61 YE= 11 DYY= 1
 DX= 60 DY= 10 E= -20 DZY= 20 -80

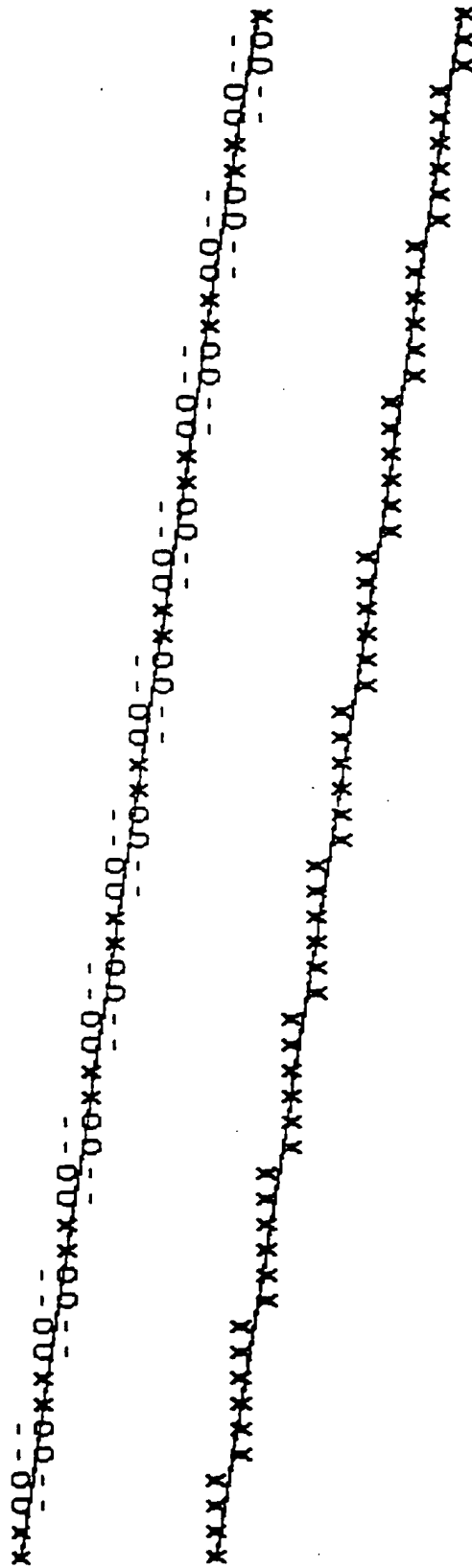


Fig. 12

XS= 1 YS= 1 XE= 62 YE= 11 DX= 61 DY= 10
XS= 1 YS= 1 XE= 62 YE= 11 DYY= 1
DX= 61 DY= 10 E= 19 DZY= 20 -42

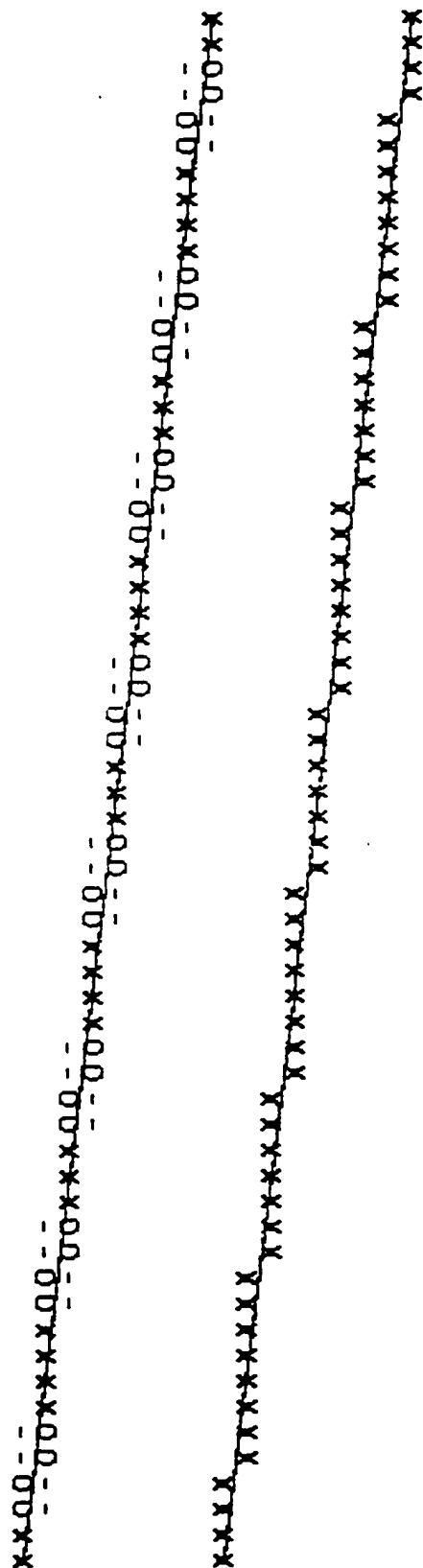


Fig. 13

XS= 1 YS= 1 XE= 61 YE= 9 DX= 60 DY= 8
 XS= 1 YS= 1 XE= 61 YE= 9 DYY= 1
 DX= 60 DY= 8 E= 4 DZY= 16 -56

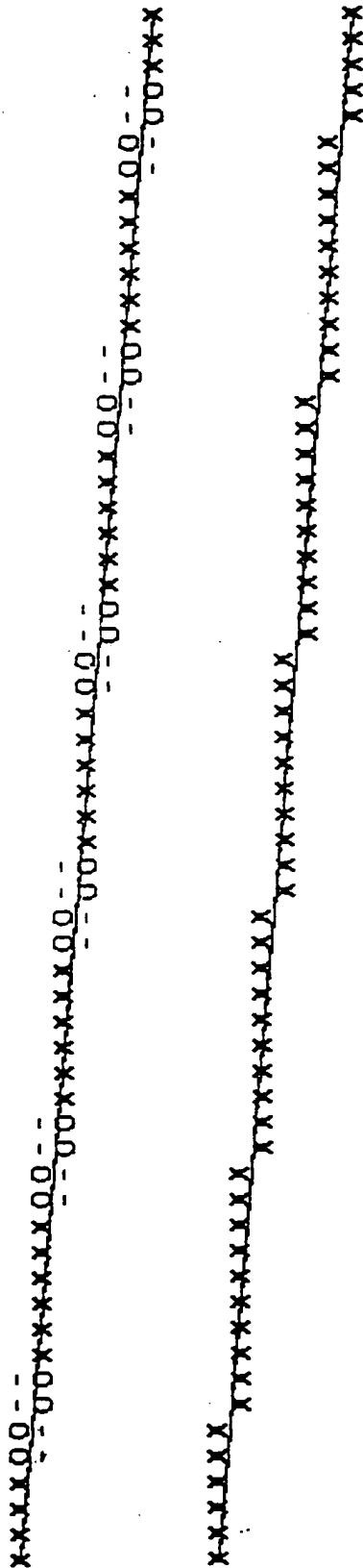


Fig. 14

XS=	1	YS=	1	XE=	61	YE=	7	DX=	60	DY=	6
XS=	1	YS=	1	XE=	61	YE=	7	DYY=	1		
DX=	60	DY=	6	E=	-12	DZY=	12		-72		

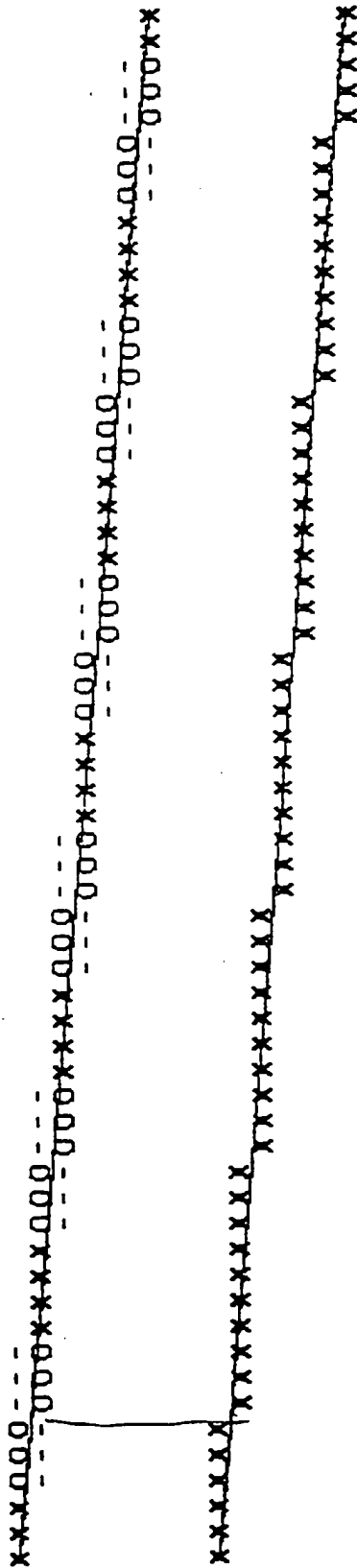


Fig. 15

XS= 1 YS= 1 XE= 62 YE= 7 DX= 61 DY= 6
XS= 1 YS= 1 XE= 62 YE= 7 DYY= 1
DX= 61 DY= 6 E= -1 DZY= 12 -50

Fig. 16

XS=	1	YS=	1	XE=	61	YE=	4	DX=	60	DY=	3
XS=	1	YS=	1	XE=	61	YE=	4	DYY=	1		
DX=	60	DY=	3	E=	-30	DZY=	6		-84		

Fig. 17

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-----0000000000000000000000000000000000000000
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

XS= 1  YS= 1  XE= 61  YE= 2  DX= 60  DY= 1
XS= 1  YS= 1  XE= 61  YE= 2  DYY= 1
DX= 60  DY= 1  E= -50  D2Y= 2  -108

```